REMARKS

This Application has been carefully reviewed in light of the Office Action mailed June 9, 2004. In order to advance prosecution of this case, Applicants amend Claims 1-3, 8, 9, 16, and 24. Applicants cancel Claims 4-7 without prejudice or disclaimer. Applicants respectfully request reconsideration and favorable action in this case.

Provisional Double Patenting Rejections

The Examiner *provisionally* rejects Claims 1-31 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over Claim 1-27 of copending Application 09/792,550 (Attorney's Docket No. 073671.0124). It is a provisional obviousness-type double patenting rejection because the conflicting claims have not been patented. Although Applicants respectfully dispute this rejection, Applicants respectfully note that amendments made to the claims of Application 09/792,550 in a recently filed Response to Office Action obviate this rejection.

Section 103 Rejections

The Examiner rejects Claims 1-31 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,421,013 issued to Smith ("*Smith*") in view of U.S. Patent No. 5,745,778 issued to Alfieri ("*Alfieri*") and EP 05 527 392 A2) to Farrell ("*Farrell*"). As amended, Claim 1 recites:

> A portable thread environment comprising:
> an application programming interface configured to support multiple application program tasks, wherein each task is either a preemptive task comprised of preemptive threads or a cooperative task comprised of cooperative threads;
> host adaptation logic for communicatively interfacing said cooperative tasks, preemptive tasks, cooperative threads and preemptive threads with a host processing environment; and
> a scheduler operable to:
> receive a request for a requested thread assigned to a first task;
> suspend the currently running thread if the requested thread is not a cooperative thread or if the currently running thread is not assigned to the first task; and
> continue running the currently running thread if the requested thread is a cooperative thread and the currently running thread is assigned to the first task.

Smith, *Alfieri*, and *Farrell*, alone and in combination, fail to disclose, teach, or suggest every element of amended Claim 1. In particular, the proposed *Smith-Alfieri-Farrell* combination fails to disclose a scheduler operable to "suspend the currently running thread if the requested thread is not a cooperative thread or if the currently running thread is not assigned to the first task." As the Examiner acknowledges, "[t]he reference of Smith fails to further explicitly teach of 'cooperative' task and 'preemptive' task[.]" *Office Action*, p.4. The Examiner fails to address *Alfieri* with respect to this element, but *Alfieri*, in fact, also fails to disclose a scheduler operable to "suspend the currently running thread if the requested thread is not a cooperative thread or if the currently running thread is not assigned to the first task," as recited by amended Claim 1.

The Examiner asserts that *Farrell*:

> teaches of thread environment where threads are organized into class. Threads from the same class are not preemptable by any thread from the same class (cooperative threads) but can be preempted by a thread from a different class (preemptive threads, see Farrell, Fig. 1, col. 2, line 33 to col. 3, line 36), for the reason to permit an application program developer to influence the order of execution, and to gain greater control by the application program developer (see Farrell, col. 2, lines 21-40).
>
> *Office Action*, p. 4.

Applicants respectfully note that, to whatever extent this description of *Farrell* may be accurate, *Farrell* does not disclose a scheduler operable to "suspend the currently running thread if the requested thread is not a cooperative thread or if the currently running thread is not assigned to the first task," as recited in amended Claim 1. None of the cited portions of *Farrell* disclose this element.

Instead, *Farrell* actually teaches away from this element, noting that "while [the highest priority] thread is dispatchable and being executed, *no other* thread from the same dispatch can preempt it unless this executing thread voluntarily relinquishes control of the processor, even if the other thread has a higher priority" because, as *Farrell* indicates, "[t]his other thread would have been created or made available after the currently executing thread was selected for the run list." Col. 2, ll. 58 - col. 3, ll. 7; emphasis added. As the M.P.E.P. states, "[a] prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention." M.P.E.P. § 2141.02, underlining in original. Thus, this contradictory portion of *Farrell* must be considered in determining

whether the prior art teaches the elements of amended Claim 1. As a result, the *Smith-Alfieri-Farrell* combination not only fails to disclose, teach, or suggest a scheduler operable to "suspend the currently running thread if the requested thread is a preemptive thread or if the requested thread is not assigned to the first task" as recited by amended Claim 1, but the proposed combination in fact clearly teaches away from the inclusion of such an element in any *Smith-Alfieri-Farrell* combination.

Consequently, the *Smith-Alfieri-Farrell* combination fails to disclose, teach, or suggest every element of amended Claim 1 and, in fact, teaches away from at least one element of Claim 1. Claim 1 is thus allowable for at least these reasons. Applicants respectfully request reconsideration and allowance of Claim 1 and its dependents.

As amended, Claim 9 recites:

> A method for porting an application from a first host environment to a second host environment, said second host environment having application-specific hardware that supports a first set of functions, and said method comprising:
>
> modeling, with a first set of tasks within said application, said functions that are supported by said application-specific hardware, wherein each task comprises one or more program fragments;
>
> removing said first set of tasks from said application;
>
> loading said application without said first set tasks to said second host environment, said application-specific hardware providing said functions provided by said first set of tasks in said first host environment; and
>
> configuring said first set of tasks and second set of tasks to communicate by passing a set of messages in said first host environment, wherein one or more of said set of messages are also used to provide communications between said first set of tasks and said application-specific hardware in said second host environment.

Smith and *Alfieri*, both alone and in combination, fail to disclose every element of Claim 9. *Smith* discloses an application programming interface for developing portable, multithreaded application programs. Abstract, ll. 1-3. *Alfieri* discloses a multi-thread processing system that balances out a workload between processors in a multiprocessor system. Abstract, ll. 1-3, 11-14.

Neither *Smith* nor *Alfieri*, however, discloses a "second host environment having application specific hardware that supports a first set of functions[.]" The Examiner fails to identify any portion of *Alfieri* that discloses a "second host environment having application specific hardware that supports a first set of functions," as recited in amended Claim 8.

Furthermore, the portion of *Smith* cited by the Examiner merely discloses an application programming interface. The application programming interface of *Smith* represents software that is "written in an object oriented language", however, not application specific hardware. Col. 2, ll. 3-4. Therefore, the proposed *Smith-Alfieri* combination fails to disclose, teach, or suggest a "second host environment having application specific hardware that supports a first set of functions[.]" As a result, the *Smith-Alfieri* combination also fails to disclose "modeling, with a first set of tasks within said application, said functions that are supported by said application-specific hardware[,]" "said application-specific hardware providing said functions provided by said first set of tasks in said first host environment[,]" and "configuring said first set of tasks and second set of tasks to communicate by passing a set of messages in said first host environment, wherein one or more of said set of messages are also used to provide communications between said first set of tasks and said application-specific hardware in said second host environment."

Consequently, the *Smith-Alfieri* combination fails to disclose, teach, or suggest every element of amended Claim 9. Claim 9 is thus allowable for at least this reason. Although of differing scope from Claim 9, Claims 16 and 24 include elements that, for reasons substantially similar to those discussed with respect to Claim 9, are not disclosed by the cited references. Claims 16 and 24 are thus allowable for at least these reasons. Applicants respectfully request reconsideration and allowance of Claims 9, 16, and 24, and their respective dependents.

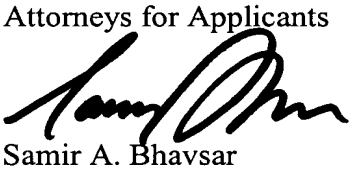
Conclusions

Applicants have made an earnest attempt to place this case in condition for allowance. For the foregoing reasons, and for other reasons clearly apparent, Applicants respectfully request full allowance of all pending Claims. If the Examiner feels that a telephone conference or an interview would advance prosecution of this Application in any manner, the undersigned attorney for Applicants stands ready to conduct such a conference at the convenience of the Examiner.

The Commissioner is hereby authorized to charge any fees or credit any overpayments to Deposit Account No. 02-0384 of Baker Botts L.L.P.

Respectfully submitted,

BAKER BOTTS L.L.P.
Attorneys for Applicants



Samir A. Bhavsar
Reg. No. 41,617

2001 Ross Avenue, Suite 600
Dallas, Texas 75201-2980
(214) 953-6581

Date: September 8, 2004

CORRESPONDENCE ADDRESS:

Customer Number: **05073**